TECHNICAL RESEARCH REPORT

Rollout Algorithms for Integrated Topology Control and Routing in Wireless Optical Backbone Networks

by Abhishek Kashyap, Kwangil Lee, Mark Shayman

TR 2003-51

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **2003**	2. REPORT TYPE	3. DATES COVERED **-**
4. TITLE AND SUBTITLE **Rollout Algorithms for Integrated Topology Control and Routing in Wireless Optical Backbone Networks**		5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Air Force Office of Scientific Research,875 North Randolph Street,Arlington,VA,22203-1768**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**12**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Rollout Algorithms for Integrated Topology Control and Routing in Wireless Optical Backbone Networks*

Abhishek Kashyap, Kwangil Lee and Mark Shayman
Department of Electrical and Computer Engineering and Institute for Systems Research
University of Maryland, College Park, MD 20742
{kashyap, kilee, shayman}@glue.umd.edu
Tel : 301-405-8843, FAX: 301-314-9281

***Abstract*--We consider a wireless backbone network with free space optical point-to-point links. Such a network could form a backbone for either a cellular or hierarchical ad hoc network. Each backbone node has a limited number of transceivers with which to establish links to neighbors. Given estimated aggregate traffic demands between source and destination backbone nodes, we consider the problem of topology control and routing--determining which links to set up and which routes to establish in order to maximize the throughput. While the problem may be formulated as an integer linear program, its solution is computationally prohibitive. Consequently, we use the mathematical technique of rollout to develop effective heuristic algorithms. Through simulation experiments, we show that the performance of the rollout algorithms we derive is clearly superior to that of the initial heuristic algorithms on which they are based.**

Keywords--Mathematical programming/optimization, Simulations

I. INTRODUCTION

World wide internet services, data communications, multimedia, virtual navigation and telemedicine are demanding greatly increased bandwidth on wireless networks. Over the last few years, a number of approaches have been taken to meet the explosive traffic growth. They include efficient signal coding and modulation schemes, spatial processing using microwave phased array antennas, and the transfer to higher radio frequency for the carrier [1]. More recently, free-space optics is attracting great attention as an alternative to radio because of its attractive characteristics. Free-space optics technology is expected to deliver unprecedented bandwidth, massive carrier reuse, ultra-low interchannel interference, low power consumption, and cost savings where electrical wires and optical fibers are too expensive to deploy and maintain. A key distinguishing feature of optical wireless is that the links are point-to-point rather than broadcast. Also, it has wide applicability from long range satellite to indoor wireless communications [1]. Therefore wireless communication network design using free space optics has become an important issue.

Two of the major issues in wireless communication networks are topology control and routing. The purpose of topology control is to select neighbors with which to establish links in order to communicate optimally with other nodes. Most research for topology control so far has focused on RF (radio frequency) networks. (See e.g., [4, 5, 11-15]. In RF-wireless networks with isotropic antennas, topology control is closely related to power control. Power is controlled to reduce the transmission range to conserve power and decrease interference while providing adequate connectivity.

The problem of topology control for optical wireless networks is different since the links are point-to-point as opposed to broadcast. In optical wireless networks, each node has a limited number of transceivers--and hence can establish links with only a limited number of nodes within its transmission range. Thus, topology control is concerned with determining the neighbors with which to establish the limited number of possible links. The recent work [3] investigates topology control for optical wireless networks under the restriction that the topology must be a ring.

There are important differences between topology design for reconfigurable wireline optical networks and topology control for wireless optical networks. In the wireline case, transmission range (lightpath length) is not a major issue. Furthermore, if the optical layer has

* Research partially supported by the Air Force Office of Scientific Research under contract F496200210217.

sufficient resources so the routing and wavelength assignment problem is always solvable, then whenever a source and destination both have available interfaces, a direct connection (one logical hop) can be established. In contrast, in the wireless case, unless the destination is within the transmission range of the source, a multihop connection is required. For these reasons, the many published results on logical topology design for wireline optical networks [6-9] are not directly applicable to free space optical networks.

Routing deals with mapping the traffic into the topology. The objective of routing is to provide paths in order to guarantee the QoS requested by the services and/or to use network resources efficiently. This can be achieved to some extent by choosing a good QoS routing algorithm. However, sometimes it is not possible to provide the requested QoS by any choice of since there are no paths available in the given topology. These situations can be mitigated by changing the topology dynamically in order to meet the QoS of the services. However, most research considers routing as separate from topology control.

In this paper, we consider the topology control and routing issues in wireless networks with point-to-point links--e.g., free space optical networks. Our objective is to control the topology in order to optimize the routing and thus to maximize the network throughput. We make no restrictive assumptions on the type of topology produced.

The algorithms we develop are specifically designed for networks with optical wireless *backbones*. Such a network could be a cellular radio network in which the base stations are interconnected by free space optical links. The optical wireless backbone consists of the base stations together with additional switching nodes. Alternatively, the free space optical backbone could be used to interconnect ad hoc networks.

The specific problem we consider is as follows: We have a geographically distributed set of nodes which are either stationary or have limited mobility. Each node has a limited number of wireless optical transmitters and receivers. A subset of nodes constitutes sources and destinations. An estimate of the aggregate traffic demand between each source and destination is available. The goal is to determine the topology and route the traffic demands so as to maximize the overall throughput of the network. In the cellular model, the demand between a source and destination pair represents the aggregate demand between the cells associated with the source and destination backbone nodes. The topology and routes are computed offline by a centralized server, and the topology is set up as computed. When the individual flows arrive, they are routed according to the routes computed for that source-destination pair in the offline phase.

We formulate the topology control and routing problem for free space optical networks as an Integer Linear Program (ILP). Since the solution of the ILP is NP-complete, we develop heuristic algorithms to provide good suboptimal solutions. The mathematical technique we employ is called *rollout* [10]. Rollout is a general technique for obtaining good solutions to Markov decision processes by systematically improving on a base (initial) heuristic. In many cases, rollout has been found to produce near optimal solutions. Rollout can be specialized to multistage discrete optimization problems. We show how the topology control and routing problem can be formulated as such a problem and develop several rollout algorithms starting with a reasonable heuristic based on shortest paths. Through simulation experiments, we are able to show that the best of the rollout algorithms significantly outperforms the shortest path heuristic.

This paper is organized as follows. Section 2 describes the topology control and routing problem using integer linear programming. Section 3 proposes an integrated topology control and routing algorithm using virtual graph. We propose several rollout algorithms in order to get sub-optimal solutions for our integrated algorithm in Section 4. And, in Section 5, we analyze the performance of the proposed algorithms using various simulations. Finally, we conclude the paper in Section 6.

II. PROBLEM DEFINITION

In this section, we describe the problem of topology control and routing in point-to-point wireless backbone networks. We formulate the problem as an Integer Linear Program (ILP) in this section.

A. Network Model

We consider wireless backbone networks. So, each wireless node in this paper refers to a backbone wireless node and is equipped with point-to-point wireless optical interfaces. By the term ‘node’ we implicitly mean “backbone node”. Each node has the capability to perform routing. And, we assume that it does not move very frequently. We also assume that wireless links can be set up in any direction with all the nodes within the transmission range. We do not consider the possibility of

optical beam obscuration, but this assumption is not essential. Since the transmission distance is related to the power level of the node, the power level and thus the transmission range of each node can be different. The wireless links are unidirectional. If there is a pair of unidirectional links between two nodes, the link capacities may differ. The number of transmitters and receivers at each node is limited, thereby restricting the number of nodes to which it can connect.

We have a traffic matrix, which consists of the aggregate traffic demands between the sources and destinations. We assume that each node in the network can be a source and/or destination. The traffic demand from node x to node y can be different from the traffic demand from node y to node x. The routing algorithms find only a single path for each demand. So we block a demand when there is no single path that can accommodate the traffic demand.

All network information and traffic information is gathered by a centralized server. The centralized server has to compute all topologies and routes using the information provided by each node. This is done offline by the server, which is assumed to have a large computing power. Then, it distributes the information to all nodes. We assume that a communication channel is available from the server to all nodes for this purpose. The server should recompute the topology and routes whenever either the traffic matrix or the (backbone) node locations change significantly. We do not anticipate that this would be done more often than hourly.

B. *Topology Control and Routing Problem*

In the problem, the following will be given as input.

- V — A set of wireless backbone nodes. Each node has a unique identification number.
- TR_i, RR_i — The number of transmitters and receivers at node i.
- C_{ij} — The channel capacity. C_{ij} indicates the bandwidth associated with a wireless link from node i to node j.
- r_i — Transmission range. The maximum distance the signal can reach from node i without any bandwidth degradation.
- T — Traffic matrix. Each entry t_{sd} represents the aggregated traffic demand from source s to destination d. Given some ordering of the traffic matrix, we can also refer to its entries as t_i, where the source-destination pair corresponding to the entry is mapped to the index i.

The algorithm produces the following output:

- E_{ij} — Topology binary variable. E_{ij} is 1 if a wireless link from node i to node j is set up, 0 otherwise.
- Y_{ij}^{sd} — Routing binary variable. Y_{ij}^{sd} is 1 if the route for t_{sd} passes through link E_{ij}, 0 otherwise.
- β_{sd} — Blocking binary variable. β_{sd} is 1 if t_{sd} gets a route with sufficient bandwidth in the network, 0 otherwise (if the demand t_{sd} is blocked).

1) *Constraints*

Before we describe constraints for topology control and routing, we define some functions used in this paper. Let *dist (i,j)* be a function that computes the distance between node i and node j. The topology control and routing algorithm should satisfy the following conditions:

$$dist(i,j) \le r_i, if \;\; E_{ij} = 1. \tag{1}$$

$$\sum_j E_{ij} \le TR_i, \forall \; i. \tag{2}$$

$$\sum_i E_{ij} \le RR_j, \forall \; j. \tag{3}$$

$$\sum_{s,d} t_{sd} Y_{ij}^{sd} \le C_{ij} E_{ij}, \forall \; i, j. \tag{4}$$

$$\sum_v Y_{sv}^{sd} = \beta_{sd}, \forall \; s, d. \tag{5}$$

$$\sum_v Y_{vs}^{sd} = 0 \;, \forall \; s, d. \tag{6}$$

$$\sum_v Y_{dv}^{sd} = 0 \;, \forall \; s, d. \tag{7}$$

$$\sum_v Y_{vd}^{sd} = \beta_{sd}, \forall \; s, d. \tag{8}$$

$$\sum_i Y_{iv}^{sd} = \sum_j Y_{vj}^{sd}, \forall \; v \ne s, d. \tag{9}$$

- Equations (1) through (3) are related to topology control. Equation (1) says that neighbors of a node should be within its transmission range. Equations (2) and (3) state that the number of links from/to a node should not exceed the number of transmitters/receivers.
- Equation (4) is link capacity constraint. The aggregate flow passing through a wireless link should not exceed the capacity of that link.
- Equations (5)-(9) state that a path contains an outgoing link at its source node, an incoming link at its destination node, and exactly one incoming and outgoing link at each intermediate node. If necessary, we eliminate any loops.

2) *Objective Function*

Our objective function is to maximize the network throughput. The throughput is defined as the sum of all traffic demands that can be successfully routed in the network as shown in (10).

$$\text{Maximize:} \quad \sum_{s,d} \beta_{sd} \times t_{sd}\,. \tag{10}$$

III. INTEGRATED TOPOLOGY CONTROL AND ROUTING

The ILP problem as described above is NP-complete. So, we need to come up with heuristic algorithms to get a suboptimal solution in polynomial time. Before explaining our approach, we introduce a simple graph model to design topology and to choose routes for flows in multi-hop wireless networks.

A. *Virtual Graph*

Given a potential topology (which we call a virtual graph), topology control is to select TR neighbors a node can transmit to and RR neighbors a node can receive from, for all the nodes in the virtual graph. Let B_i denote a set of nodes that V_i can connect to i.e., the potential neighbors of node V_i. We call the virtual links between V_i and each node in B_i potential links. We call the graph containing all nodes and all the potential links the Virtual Graph (VG). Let us consider an example demonstrating how we can construct a virtual graph from a given wireless network shown in Fig. 1. For simplicity in the illustration of concepts, we assume that a node can establish only bidirectional links with its neighbors in all the examples, though we actually work with unidirectional links in our algorithms. The weight of each link in the figure represents the distance between the two nodes it connects. We assume the transmission range of each node is 1. In the figure, we do not show the links with weight much greater than 1 for simplicity and ease of understanding.

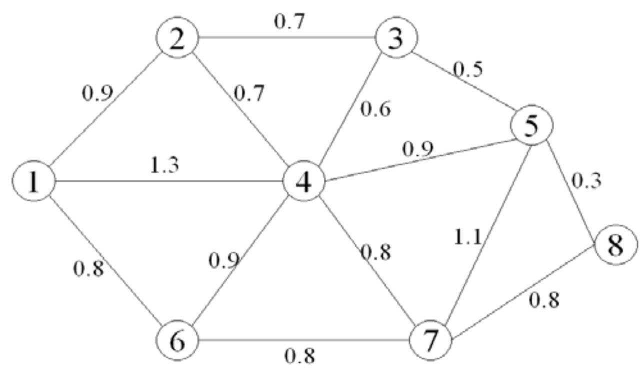


Figure 1. Example Wireless Network

We eliminate all links in Fig. 1 which violate the distance constraint of (1). The resulting graph is the Virtual Graph, which is shown in Fig. 2. The virtual graph gives all possible links we have to consider for topology control. So, the topology control problem can be defined as a node (link) deletion problem among all neighbors (links) until (2) and (3) are satisfied for all nodes in the virtual graph.

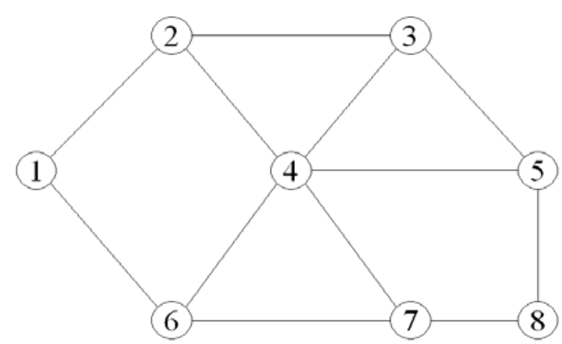


Figure 2. Example Virtual Graph

B. *Integrated Topology Control and Routing*

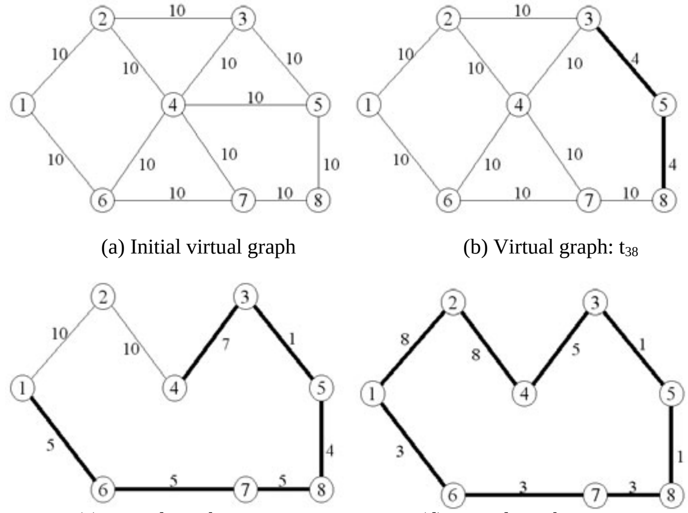
In this section, we propose a heuristic approach for topology control and routing in order to maximize our

objective function. The approach is to integrate topology control and routing, i.e., the decision of selecting the wireless link at each node and of finding a route for the aggregate flows is made simultaneously. This is enabled by weighting the edges of the virtual graph with available link bandwidth. The following steps are taken for computing the topology and routes for the demands given in the aggregate traffic matrix:

1. A demand is chosen based on some criteria and a locally optimal path (satisfying the interface constraints and bandwidth constraints) is computed for the demand. If none exists, then the demand is rejected.
2. If the path includes potential links, then those links are marked as actual links.
3. The capacity of each link on the path in the virtual graph is updated (decreased) to incorporate the bandwidth allocated to the demand routed.
4. The virtual graph is updated by eliminating all the potential links that lead to the violation of interface constraints of (2) and (3).
5. Steps 2, 3, 4 and 5 are repeated until all demands are either provisioned or rejected. This way, a topology is created from the virtual graph and all the routes are computed for the demands given in the traffic matrix (the ones we are able to route, the others are rejected).

Let us explain this approach of integrated topology control and routing with an example. In this example, we assume that each node has two interfaces available for establishing bidirectional links. The traffic matrix is sorted in the order of decreasing demands, and demands are selected in that order. The link capacity of each link is assumed to be 10 units. We use constraint based shortest-path routing for path selection, with the constraint being that of the interface limit and the weight of each link assumed to be 1. Let the traffic demands be: t_{38}= 6, t_{18}= 5, t_{45}= 3, t_{37}= 2. We can compute the shortest path for the first demand (first entry of the traffic matrix) t_{38} using the graph as shown in Fig. 3(a). The shortest path for the traffic demand t_{38} is 3-5-8.

Fig. 3(b) shows the virtual graph after converting the potential links along the path 3-5-8 to actual links and allocating the bandwidth for the demand. In Fig. 3(b), the actual (wireless) links are represented by thick lines and the potential links are represented by thin lines. As the number of available interfaces per node is two and node 5 uses those interfaces for links with node 3 and node 8, there are no more interfaces available for node 5 to establish a link with other nodes. Thus, the potential link between node 4 and 5 is eliminated, as can be seen by comparing Figures 3(a) and 3(b).



(a) Initial virtual graph (b) Virtual graph: t_{38}

(c) Virtual graph: t_{38},t_{18},t_{45} (d) Virtual graph : t_{38},t_{18},t_{45},t_{37}

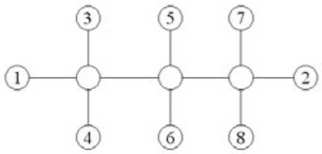
Figure 3. Virtual Graph after t_{48}

From the virtual graph in Fig. 3(b), we find the shortest path 1-6-7-8 for the demand t_{18} and the shortest path 4-3-5 for t_{45}. Fig. 3(c) shows the updated virtual graph which reflects the routing of these demands. Now we compute the shortest path for the demand t_{37} using the virtual graph of Fig. 3(c). There are two paths available for t_{37}: 3-5-8-7 and 3-4-2-1-6-7. Since the available bandwidth along the path 3-5-8-7 is 1, which is less than the demand, the path cannot be selected even though it is the shortest path in the virtual graph. So, the shortest path for the demand t_{37} is computed as 3-4-2-1-6-7, and the virtual graph updated to get the final topology as shown in Fig. 3(d).

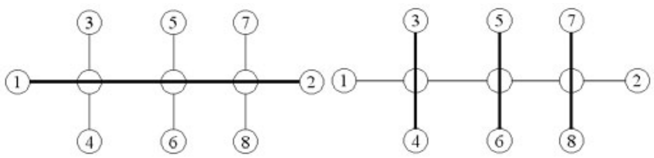
C. *Issues in Integrated Topology Control and Routing*

The purpose of our integrated approach is to maximize the network throughput while routing demands sequentially. There are two key issues related to it. Let us consider them with the help of two example networks shown in Figures 4 and 5. In these examples, the number of available interfaces at each node is two, and the links are assumed to be bidirectional for simplicity. Given the traffic matrix {t_{12},t_{34},t_{56},t_{78}}, all demands being same, consider the path provisioning and topology design for the network in Fig. 4(a). When we provision a path for t_{12} first, we get the graph as shown in Fig. 4(b) resulting in only one demand being provisioned. If we consider the other traffic demands first, then this demand cannot be provisioned but the

other three demands can be provisioned. Thus, the topology of Fig. 4(b) *ERROR*(c)* resulting from choosing the last three traffic demands before the first one gives a better throughput.



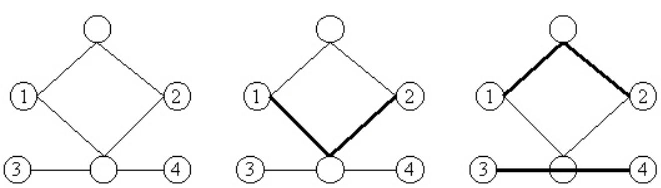
(a) Example network



(b) Path for t_{12} (c) Paths for t_{34},t_{56},t_{78}

Figure 4. Topology Generation by considering demands in different order

Let us consider another example using Fig. 5. We consider path provisioning for the sorted traffic demands $\{t_{12}, t_{34}\}$. There are two paths available for the demand t_{12}. If we choose a path for t_{12} as shown in Fig. 5(b), then we can not provide a path for t_{34} because of the interface constraint at an intermediate node. However, when we choose the other path as shown in Fig. 5(c), both the traffic demands can be provisioned.



(a) Example network (b) Path for t_{12} (c) Paths for t_{12}, t_{34}

Figure 5. Topology Generation by using different shortest paths for a demand

The above examples illustrate the importance of the two factors that affect the network throughput in our integrated algorithm: The sequence in which we route the demands given in the traffic matrix, and the selection of paths for routing the demands. These two factors have an effect on which potential links will be deleted from the virtual graph because of the interface constraints and the capacity of which links will be decreased by the amount of the routed demand. So, these factors affect the future path computations and the topology control. Since finding the optimal solution as explained in section 2 is an NP-complete problem, we consider suboptimal algorithms to take care of these in the next section.

IV. ROLLOUT ALGORITHMS FOR TOPOLOGY CONTROL AND ROUTING

As mentioned in the previous section, the performance of the topology control and routing depends on the order in which the traffic demands are considered for link formation and routing, and the selection of the path for each demand. We start with reasonable heuristics for demand ordering and path selection and use the rollout technique to improve the heuristics to obtain potentially near-optimal solutions.

A. *Basic Rollout Algorithm*

Rollout is a general method for obtaining an improved policy for a Markov decision process starting with a base heuristic policy [10]. The rollout policy is a one step look-ahead policy, with the optimal cost-to-go approximated by the cost-to-go of the base policy. We use the specialization of rollout to discrete multistage deterministic optimization problems. Consider the problem of maximizing $G(u)$ over a finite set of feasible solutions U. Each solution u consists of N components $u = (u_1, \ldots, u_N)$. We can think of the process of solving this problem as a multistage decision problem in which we choose one component of the solution at a time. Suppose that we have a heuristic algorithm, the so-called "base heuristic", that given a partial solution $(u_1, \ldots, u_n)$ $(n < N)$ extends it to a complete solution $(u_1, \ldots, u_N)$. Let $H(u_1, \ldots, u_n) = G(u_1, \ldots, u_N)$. In other words, the value of H on the partial solution is the value of G on the full solution resulting from application of the base heuristic. The rollout algorithm R takes a partial solution $(u_1, \ldots, u_{n-1})$ and extends it by one component to $R(u_1, \ldots, u_{n-1}) = (u_1, \ldots, u_n)$ where u_n is chosen to maximize $H(u_1, \ldots, u_n)$. Thus, the rollout algorithm considers all admissible choices for the next component of the solution and chooses the one that leads to the largest value of the objective function if the remaining components are selected according to the base heuristic. It can be shown that under reasonable conditions, the rollout algorithm will produce a solution whose value is at least as great as the solution produced by the base heuristic. The rollout algorithm typically achieves a substantial performance improvement over the base heuristic at the expense of extra computation that is equal to the computation time of the base heuristic times a factor that increases

polynomially with the problem size.

B. Rollout Algorithms for Topology Control and Routing

We propose rollout algorithms for topology control and routing in this paper. In this section, we propose four different rollout algorithms: index rollout, route rollout, sequential rollout and integrated rollout.

1) Index Rollout Algorithm

The example in Fig. 4 shows that the order in which traffic demands are routed plays an important role in determining the throughput of the resulting topology. Index rollout seeks to optimize this order. The base heuristic works as follows: Suppose that a partial topology has been obtained by choosing routes for n demands (t_1, …, t_n) from the traffic matrix. The base heuristic routes the remaining demands in decreasing order of magnitude. (Routing demands in decreasing order of magnitude is known to be a useful heuristic for reconfigurable wireline optical networks [7, 9]). For each demand, it chooses a route using constrained shortest path first (CSPF). Thus, t_{n+1} is the largest remaining demand. The route chosen for this demand is a shortest unidirectional path in the partial topology satisfying the constraints. This means that every actual link in the path must have sufficient residual bandwidth for the demand; every virtual link in the path must have an available transmitter at its head node and an available receiver at its tail node. If there is no feasible path, then the 'null' route is assigned--i.e., the demand is blocked. Once t_{n+1} has been routed, the base heuristic routes the next largest demand t_{n+2} in the same way using the partial topology existing after t_{n+1} has been routed. The base heuristic algorithm continues in this way until all demands have been routed (or assigned null routes).

The index rollout algorithm works as follows: In the first step, the rollout algorithm uses CSPF to route the demand t_1 determined by the requirement that it maximize the total network throughput when the base heuristic is used to complete the topology starting with t_1. Now, suppose that the demands (t_1, …, t_{n-1}) have been routed in this order by the rollout algorithm. In the next step, the rollout algorithm uses CSPF to route the remaining demand t_n determined by the requirement that it maximize the total network throughput when the base heuristic is used to complete the topology starting with (t_1, …, t_n). In other words, routing t_n next minimizes the sum of the remaining demands that are blocked.

2) Route Rollout Algorithm

The example in Fig. 5 shows that the choice of path for each traffic demand plays an important role in determining the throughput of the resulting topology. Route rollout seeks to optimize the selection of path for each demand *when the demands are considered in a fixed order.* We consider the demands in decreasing order of magnitude. (Additional algorithms may be obtained by using different criteria to order the traffic demands; see Section 4.2.3 below.) Let (t_1, …, t_N) be the ordered sequence of demands. The base heuristic works as follows: Suppose that a partial topology has been obtained by choosing routes (p_1, …, p_n) for the first n demands (t_1, … , t_n). The base heuristic routes the remaining demands (t_{n+1}, …, t_N) sequentially using CSPF.

The route rollout algorithm works as follows: Fix an integer K > 1. In the first step, the rollout algorithm considers at most K feasible shortest paths as candidates for the route p_1 for the demand t_1. For each potential choice of p_1 it uses the base heuristic to complete the topology by routing the remaining traffic demands. The rollout algorithm then selects for p_1 the candidate that results in the maximum total network throughput. Now, suppose that the demands (t_1, …, t_{n-1}) have been given routes (p_1, …, p_{n-1}) by the rollout algorithm. In the next step, the rollout algorithm considers at most K feasible shortest paths as candidates for the route p_n for the demand t_n. For each potential choice of p_n it uses the base heuristic to complete the topology by routing the remaining traffic demands. The rollout algorithm then selects for p_n the candidate that results in the maximum total network throughput. Note that if there is only one feasible shortest path for a traffic demand, the routing decision made by the rollout algorithm coincides with the decision made by the base heuristic.

It might appear desirable to consider all feasible shortest paths as candidates for p_n. However, this is not possible since the problem of finding all such paths requires exponential time. Consequently, we limit the number of paths considered to K, where the upper bound K is chosen small enough to allow reasonable computation time given the size of the network.

3) Sequential Rollout Algorithm

Thus far, we have considered rollout algorithms either for the sequence of traffic demands or for path selection. Another possibility is to apply rollout in order to optimize both the sequence of traffic demands and the route path selection. This can be achieved by applying rollout algorithms sequentially. It means that we first apply the index rollout algorithm in order to optimize the

sequence of traffic demands as explained in 4.B.1. Then we apply the route rollout algorithm described in 4.B.2 in order to optimize the path selections for the sequence of traffic demands determined by the index rollout. The difference between the sequential rollout and route rollout algorithm is that the sequential rollout uses the sequence of traffic demands determined by index rollout while route rollout sequences the traffic demands in order of decreasing magnitude.

4) *Integrated Rollout Algorithm*

Instead of first choosing the sequence of traffic demands and then choosing the paths for the traffic demands, an alternative is to make those decisions at the same time. We call this the integrated rollout algorithm.

In integrated rollout, each component of a solution is a pair (t_k,p_k) consisting of a traffic demand and its path. Thus, the algorithm seeks to optimize the sequence $((t_1,p_1), \ldots, (t_N,p_N))$. The base heuristic takes a partial solution $((t_1,p_1),\ldots,(t_n,p_n))$ and extends it to a complete solution by choosing the remaining traffic demands $(t_{n+1}, \ldots, t_N)$ in order of decreasing magnitude and choosing paths $(p_{n+1}, \ldots, p_N)$ (some of which may be null) for these traffic demands sequentially using CSPF.

The integrated rollout algorithm works as follows: In the first step it considers pairs (t_1,p_1) where t_1 is any of the traffic demands and p_1 is any one of a maximum of K feasible shortest paths for t_1. It selects the pair (t_1,p_1) that gives the maximum total network throughput when the base heuristic is used to extend it to a full topology. Now, if the rollout algorithm has produced the sequence $((t_1,p_1), \ldots, (t_{n-1},p_{n-1}))$, it considers pairs (t_n,p_n) where t_n is a remaining demand and p_n is any one of a maximum of K feasible shortest paths for t_n. It selects the pair (t_n,p_n) that maximizes the total network throughput when the base heuristic is used to extend $((t_1,p_1), \ldots, (t_n,p_n))$ to a full solution.

V. ANALYSIS AND SIMULATION RESULTS

A. *Time Complexity Analysis*

Let the number of nodes in the network be N and the number of aggregate demands in the traffic matrix be M. We use a modified version of Dijsktra's shortest path algorithm as a heuristic for finding the shortest paths. It is modified to take care of the interface constraints while finding a shortest path. As the virtual graph and the graph at any intermediate state is not sparse, so the process of finding a shortest path takes $O(N^2)$ time. The heuristic we use for sorting is sorting by decreasing order of traffic demands, which takes O(MlogM) time for sorting M aggregate flows. This time is insignificant compared to the time taken by other components of the algorithms, so it does not show up in the time complexity of any of our algorithms.

The time complexity of the heuristic algorithm is $O(MN^2)$, as shortest paths are computed M number of times. If the set of source/destination nodes is fixed, then so is the number of aggregate demands. In this case, the complexity becomes $O(N^2)$.

The time complexity of the route rollout algorithm is $O(M^2N^2)$, as K is fixed. This complexity is due to the fact that at each decision step, O(M) shortest paths are computed, and there are M decision steps in the algorithm. In the case of fixed M, the complexity is $O(N^2)$.

The time complexity for the index rollout algorithm is $O(M^3N^2)$. At each decision step in the algorithm, $O(M^2)$ shortest paths are computed, and there are M decision steps in the algorithm resulting in the above complexity. This also reduces to $O(N^2)$ for fixed M. The complexity for the integrated rollout is also the same as the time is scaled by K which is a constant.

The time complexity for the sequential rollout is the sum of the complexity for the index and route rollout algorithms i.e., $O(M^2N^2 + M^3N^2)$ which is the same as $O(M^3N^2)$. As in the previous cases, this also reduces to $O(N^2)$ for fixed M.

In the case where each node in the network can be a source or a destination, M scales as N^2 and the complexity of the heuristic algorithm becomes $O(N^4)$, while the route rollout algorithm takes $O(N^6)$, and the other three rollout algorithms take $O(N^8)$ time.

B. *Simulation Results and Discussion*

The simulations were done with two types of network data. The first set of simulations was done with a fixed number of sources and destinations in the network. The second set of simulations was done assuming any node can be a source or a destination node.

1) *Simulation Set 1*

The network was assumed to have the following parameters:

- Number of nodes in the network = 50
- Nodes are uniformly distributed, with each node having an average of 7.5 potential neighbors.
- The transmission range of all nodes is assumed to be the same.
- Capacity of each link = 100 in each direction.

- Number of nodes capable of being a source/destination = 12.
- Number of source-destination pairs = 125, selected from among the nodes which can be sources or destinations. In this case, nearly all possible source-destination pairs are a part of the traffic matrix.
- Aggregate traffic between each pair: Uniformly distributed between 1 and 40 units.
- Number of receive interfaces at each node = 3
- Number of transmit interfaces at each node = 3
- Number of Shortest Paths considered in Route Rollout, Sequential Rollout and Integrated Rollout, K = 4
- Weight of each link for constrained shortest path computation = 1, thus making the shortest path as the constrained min-hop path.

The simulation was run 10 times and in each simulation, the network topology was formed starting with these parameters. The throughput from the aggregate traffic matrix and number of rejects (the demands which we could not route) were noted. Fig. 6 shows the throughput for 5 of the 10 simulations, and Fig. 7 shows the number of rejects for those 5 simulations. The simulations shown in these figures have been selected to show the general trend and the variation encountered in the results. Note that rejects and throughput are not directly related to each other--i.e., it is possible (but unlikely) to simultaneously achieve higher throughput and higher rate of rejection since the size of the demands is not constant.

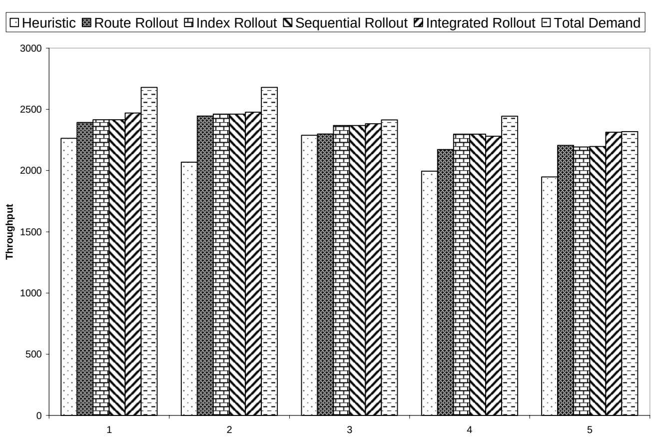


Figure 6. Throughput for Simulation Set 1

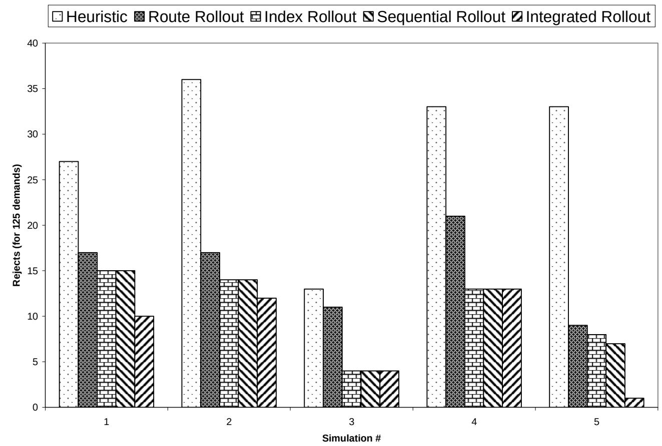


Figure 7. Rejects for Simulation Set 1

As can be seen from the figures, all the four rollout algorithms work much better than the heuristic. The integrated rollout normally works the best among these, followed by the sequential rollout, index rollout, and route rollout, in that order. There are some exceptions to the general trend, as can be seen from simulations 4 and 5. In simulation 4, the index and sequential rollouts work better than the integrated rollout and in simulation 5, the route rollout works better than the index and sequential rollouts. As all the policies are suboptimal, so none of the rollout policies is guaranteed to perform better than the others as the decision at any stage of the algorithms is not optimal. This validates the results seen in simulations 4 and 5.

Table 1 gives the average rejects over 125 aggregate demands (as a percentage of total demands) and the average throughput (as a percentage of total requested demand) over all simulations of this set.

TABLE I. AGGREGATE RESULTS FOR SIMULATION SET 1

Policy	**Throughput**	**Rejects**
Heuristic	85.13%	20.72%
Route Rollout	92.18%	10.64%
Index Rollout	94.49%	7.12%
Sequential Rollout	94.51%	7.04%
Integrated Rollout	95.16%	6.24%

As can be seen from the table, comparing with the heuristic in terms of throughput, the route rollout performs nearly 8.3% better, the index rollout performs

11% better, the sequential rollout performs 11% better and the integrated rollout performs 11.8% better. In terms of the number of rejects, the route rollout performs nearly 48.6% better, the index rollout performs 65.6% better, the sequential rollout performs 66% better and the integrated rollout performs 69.9% better than the heuristic. So, generally the integrated rollout is expected to perform the best among these rollouts.

Another observation from the results is that the index selection is more critical than the selection of routes from among multiple routes. This can be inferred from the fact that the index and sequential rollouts work much better than the route rollout while the integrated rollout does not work that much better than the index and sequential rollouts. This conclusion is further strengthened by the observation that index and sequential rollouts perform either the same or sequential rollout does slightly better than the index rollout; there is not a big margin between them, as can be seen from Table 1.

Regarding the connectivity of the network, the optimization of the throughput ensures with high probability that the source and destination nodes are all connected. If certain other nodes are not essential as transit nodes, it is possible that these nodes may be disconnected.

2) Simulation Set 2

This simulation set is for the case where all the nodes can be sources/destinations, and the network is more heavily loaded than in simulation set 1. The network was assumed to have the following parameters different from the simulation set 1:

- Number of nodes in the network = 20
- Nodes are uniformly distributed, with each node having an average of 6.5 potential neighbors.
- Any node can be a source or a destination
- Number of source-destination pairs in the traffic matrix: between 135 and 170, selected uniformly from among all possible source-destination pairs (380 of them). Relative to the size of the network, the total demand is very large compared to the network in simulation set 1. The demand for simulation set 1 is around 2500 units for a network of size 50, while it is around 2000 units for a network of size 20 here.
- Aggregate traffic between each pair: Uniformly distributed between 1 and 30 units.

The simulation was run 10 times and in each simulation, the network topology was formed starting with these parameters. Fig. 8 shows the throughput for 5 of the 10 simulations, and Fig. 9 shows the number of rejects for those 5 simulations. The simulations shown in these figures have been selected to show the general trend and the variation encountered in the results.

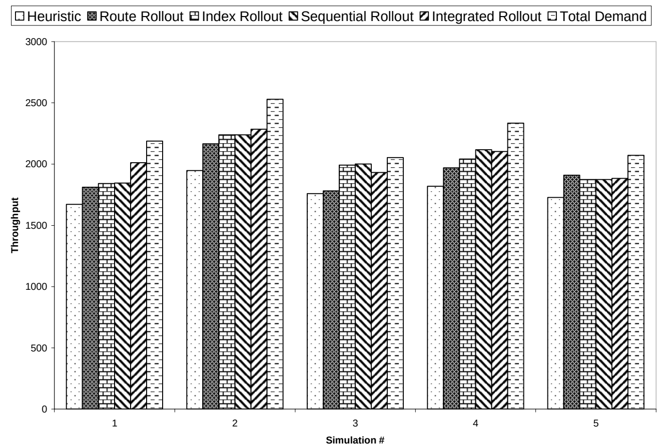


Figure 8. Throughput for Simulation Set 2

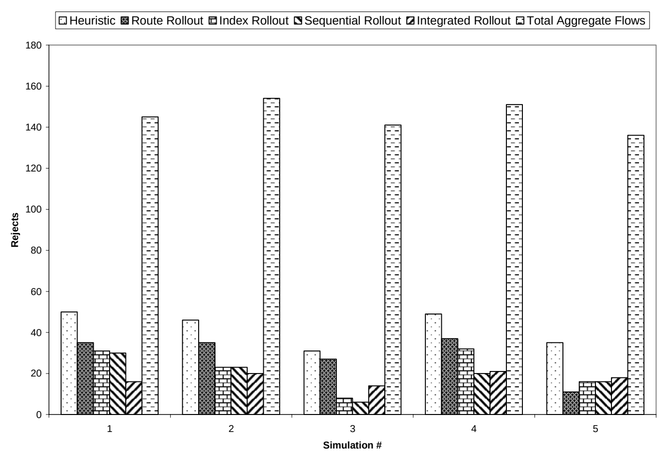


Figure 9. Rejects for Simulation Set 2

As can be seen from the figures, all four of the rollout algorithms work much better than the heuristic. The integrated rollout normally works the best among these, followed by the sequential and index rollouts, which work the same most of the time, followed by the route rollout. As in simulation set 1, there are instances when the index and sequential rollout work better than the integrated rollout.

Table 2 gives the average rejects (as a percentage of total requested aggregate flows) and the average

throughput (as a percentage of total requested demand) over all simulations of this set.

TABLE II. AGGREGATE RESULTS FOR SIMULATION SET 2

Policy	Throughput	Rejects
Heuristic	79.87%	30.56%
Route Rollout	86.59%	21.16%
Index Rollout	89.86%	14.44%
Sequential Rollout	90.25%	13.43%
Integrated Rollout	92.12%	10.61%

As can be seen from the table, comparing with the heuristic in terms of throughput, the route rollout performs nearly 8.4% better, the index rollout performs 12.5% better, the sequential rollout performs 13% better and the integrated rollout performs 15.3% better. In terms of the number of rejects, the route rollout performs nearly 30.8% better, the index rollout performs nearly 52.7% better, sequential rollout performs 56.1% better and the integrated rollout performs 65.3% better than the heuristic.

In this case also, the network was connected for each simulation as the traffic matrix was comprehensive in terms of the nodes it covered.

VI. CONCLUSION

In this paper, we have developed algorithms for integrated topology control and routing of wireless optical backbone networks. Given estimates of the aggregate traffic demands between source and destination nodes, the algorithms determine which wireless links to establish and which routes to use. The algorithms are derived using the mathematical technique of rollout. By starting with heuristic algorithms for ordering the demands in the traffic matrix and for choosing routes, rollout is applied to obtain significantly improved algorithms. Different algorithms are obtained by applying rollout to the ordering of the traffic demands, to the choice of routes for individual traffic demands, or to a combination of both.

We have done extensive simulation experiments to evaluate the performance of our algorithms. For the experiments on a 50-node network, the rollout algorithms provided as much as a 12% improvement in throughput and 70% reduction in blocked demands compared to the initial heuristic algorithm from which they were derived.

REFERENCES

[1] N. A. Riza, “Reconfigurable Optical Wireless”, *LEOS '99*. IEEE , vol.1 , pp.70-71, 8-11 Nov. 1999.

[2] Z. Yaqoob, N. A. Riza, “Smart Free-Space Optical Interconnects and Communication Links using Agile WDM Transmitters”, *2001 Digest of the LEOS Summer Topical Meetings* , 30 July-1 Aug. 2001.

[3] A. Desai, “Dynamic Topology Control of Free Space Optical Networks,” M.S. thesis, Department of Electrical Engineering, University of Maryland, 2003.

[4] R. Ramanathan and R. Rosales-Hain, “Topology Control of Multihop Wireless Networks using Transmit Power Adjustment, ” *IEEE Infocom 2000*, vol. 2, pp. 404-413, 26-30 March 2000.

[5] Z. Huang, C-C. Shen, C. Srisathapornphat and C. Jaikaeo, “Topology Control for Ad Hoc Networks with Directional Antennas, ” *ICCCN 2002*, pp. 16-21, Miami, Florida, October 2002.

[6] D. Banerjee, B. Mukherjee, “Wavelength-routed Optical Networks: Linear Formulation, Resource Budgeting Tradeoffs and a Reconfiguration Study”, IEEE/ACM Trans. Networking, vol. 8, pp. 598-607, Oct. 2000.

[7] K. H. Liu, C. Liu, J. L. Pastor, A. Roy, J. Y. Wei, “Performance and Testbed Study of Topology Reconfiguration in IP over WDM”, IEEE Transactions on Communications, in press, 2002.

[8] E. Leonardi, M. Mellia, M. A. Marsan, “Algorithms for the Logical Topology Design in WDM All-Optical Networks”, Optical Networks Magazine, pp. 35- 46, Jan. 2000.

[9] R. Ramaswami, K. N. Sivarajan, “Design of Logical Topologies for Wavelength-Routed optical Networks”, IEEE JSAC, pp. 840-851, Jun. 1996.

[10] D. P. Bertsekas, “Dynamic Programming and Optimal Control”, vol. 1, Athena Scientific, 2000.

[11] L. Li, J. Halpern, P. Bahl, Y-M. Wang and R. Wattenhofer, “Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks, ” *ACM Symposium on Principles of Distributed Computing*, 2001.

[12] V. Rodoplu and T. Meng, “Minimum Energy Mobile Wireless Networks, ” *IEEE International Conference on Communication 1998*, vol. 3, pp. 1633-1639, 7-11 June 1998.

[13] R. Wattenhofer, L. Li, P. Bahl and Y-M. Wang, “Distributed Topology Control for Wireless Ad-hoc Networks, ” *IEEE Infocom 2001*, pp. 1388-1397.

[14] S. Ruhrup, C. Schindelhauer, K. Volbert and M. Grunewald, “Performance of Distributed Algorithms for Topology Control in Wireless Networks, ” *International Parallel and Distributed Processing Symposium*, 2003.

[15] L. Bao and J. J. Garcia-Luna-Aceves, “Topology Management in Ad Hoc Networks, ” *ACM Mobihoc*, pp. 129-140, Annapolis, Maryland, June 2003.